A Data Management System for Ab-Initio Nuclear Physics Applications

Fang Liu[1], **Ritu Mundhe**[2], **Masha Sosonkina**[1], **Chase Cockrell**[3], **Miles Aronnax**[3], **Pieter Maris**[3], **James P. Vary**[3]

1**. Ames Laboratory, Iowa State University**

2**. Computer Science Department, Iowa State University**

3**. Physics Department, Iowa State University**

Keywords: data management system; computational applications; reproducible research

Abstract

Reproducible Research lies at the very core of the scientific method: an experiment is reproducible if it can be replicated by researchers independent from those that conducted it in the first place. Provenance is an emerging concept in computational sciences. In this work, we introduce provenance to another scientific domain - *ab initio* nuclear physics calculations. Through a data management system, the provenance information is consistently recorded for post-processing, validation, and education purposes. Due to the large volume of data generated through each large scale simulation, the data management system needs to address the high performance computing needs compared with a traditional database management system.

1 Introduction

One of the computational approaches in nuclear physics is the configuration interaction (CI) method for solving the many-body nuclear Hamiltonian in a sufficiently large basis space to obtain numerically converged results. A state-of-the-art CI code for *ab initio* nuclear physics has been developed by Vary and his collaborators at Iowa State University [18], and the software package Many Fermion Dynamics for nucleons (MFDn) is a part of the U.S. Department of Energy (DOE) Scientific Discovery through Advanced Computing (SciDAC)/Universal Nuclear Energy Density Functional (UNEDF) project. In MFDn, the nuclear quantum many-body Hamiltonian is evaluated in a large many-body basis constructed from Slater-Determinants of harmonic oscillator single-particle states and diagonalized by iterative techniques to obtain the low-lying eigenvalues and eigenvectors. The eigenvectors are then used to evaluate a suite of experimental quantities to test accuracy. MFDn requires a set of inputs for each run, may take several hours on up to $220,000$ cores (largest hardware we have access to) and generate large volumes of data. MFDn has good scaling properties using a combination of Message Passing Interface (MPI) and OpenMP on existing supercomputing architectures [13] and recent algorithmic improvements have significantly improved its overall performance [13, 18]. This

paper describes the first effort to document and store essential provenance information for an MFDn run and its results in a database for future retrieval. Key reasons that a result archiving system is needed include:

- Avoid repeating the same or very similar caculations to improve the efficient use of resources,
- Promote efficient and accurate research by faciliating comparison of results between alternative methods and with different input data (such as 2-body nucleon-nucleon (NN) or 3-body nucleon-nucleon-nucleon (NNN) interactions for MFDn),
- Provide a convenient way to allow students to access to state-of-the-art nuclear structure calculations.

Reproducible Research (RR) has recently captured significant attention in computer science. Provenance forms an essential part of reproducibility. Many papers describe methods of modeling [15] of provenance. Various provenance-enabled softwares and modeling techniques such as Kepler [11], PASS [15], Pegasus [8], PASOA [7], REDUX [6] have been evolved in this direction. The three main classifications of these are process-based systems, workflow-based systems, and OS-based systems. This paper is our initiative to introduce a dedicated provenance enabled Data Management System (DMS) founded on an open source relational database. The need to define RR comes from development of a ”hybrid” of theoretical and experimental research i.e. computational sciences. These science encompasses many fields, including computer science, statistics, many areas of engineering (e.g. signal processing) and pure science such as physics. RR requires a paper published to include the code and the data to reproduce all the results and a list of configurations (software version, platform). The combination of the result archiving system with the concept of RR leads us to introduce provenance via the DMS to the computational nuclear physics domain.

2 Reproducible Research

Research which is targeted towards solving a problem or reporting an innovation is archived in the form of publications. The main categories of work reported in publications are theory, physical experimentation, and numerical simulation. Theory is the only categories that is deduced completely with the help of proofs and axioms along with some assumptions and approximations. Hence it can be understood and reproduced completely by a person with sufficient theoretical and mathematical tools. The main drawback of this system for the other two categories of research are the factors that change with time and environment. This creates challenges for those who want to demonstrate the same experiment or recreate the same simulation. It also reduces the credibility of the experiment if it cannot be accurately recreated. Therefore, the concept of RR aims to account for as many variables as possible in the computational simulation.

Provenance is defined as the source or origin of an object. In laboratories, provenance of an experiment has always been stored in the form of research papers, log books and technical reports. As scientific endeavors have expanded over time, the volume, complexity and need for availability of these reports has increased dramatically. Consider a medical experiment that requires conducting hundreds of tests and analyzing results. A relevant subset of these results is to be determined and an important statement is to be made in the field of medicine. Furthermore, very practically, assume that this experiment is done by a group of people, located separately. Data must be exchanged at some point which leads to several questions that must be asked: who performed this experiment? when was it performed? Are there multiple results for the same test? how was it performed: what are all the steps, variables and environment for it? where are the observables (cell samples/x-rays/numbers) located? description of the subject? etc. A readily available list which will answer all questions exhaustively is the provenance for this experiment. In simple terms if this information is available to anybody, at any point in time, it is the application of provenance in RR.

When an experiment is performed, one has to store the event such that the experiment is easily identifiable by each and every detailed input and environmental variable used; in addition, all relevant output has to be stored such that it is easily identifiable and retrievable. For instance a scientific computational software may have variables such as operating system, kernel version, compiler used, number of nodes in a parallel computing cluster, number of processors in a node etc. Other input variables can be experiment-related values such as input interactions, total angular momentum range, numbers of neutrons/protons, etc. The complete set of these environmental and input variables are called provenance; they specify the environment necessary to reproduce the experiment. The output of the experiments have multiple formats and vary greatly in size, which may pose some challenge to a traditional DMS. The primary goal of the DMS is to make this information available and easily accessible, as per the original experimenter's discretion. The more widely available it is, the more reproducible the experiment is and the greater the impact of the research is. Credibility of research increases when an experiment is reproducible since the correctness of the results can be verified. Convenient retrieval of output from previous experiments saves the time, effort, and resources of re-running experiments, and users may move faster towards analyzing the results.

Traditionally, a record of a research study is kept by exhaustive documentation and research papers. One approach [9] is to create a portal of research papers and articles with world wide access. This portal must contain the paper, the code, specific inputs, provenance and acquired results. Since reading the paper does not lead automatically to accepting the idea represented by the paper, the user may want to run the experiment themselves and see the results. Such experimen-

tal verification often facilitates understanding and motivates future work. Further advantages and importance of RR are explained in a very convincing way in [9].

Some methods may work well in the case of papers which deal with a few experimental runs, but can quickly become cumbersome when analyzing hundreds or thousands of runs. In order to store information about an indefinite number of runs comprehensively, we use a relational database system. With this ideology, we examine the case of MFDn.

3 Overview of MFDn

In this work, we discuss the *ab initio* nuclear physics application - MFDn, developed by Vary's group at Iowa State University [18, 13]. Using a quantum many-particle framework to solve nuclear properties with 2-body and 3-body interactions is referred to as an "*ab initio*" problem and is recognized to be computationally difficult. MFDn is a massively parallel code used for large-scale nuclear structure calculations in the No-Core Shell Model (NCSM) [16, 17] and No-Core Full Configuration (NCFC) [14, 12] methods using F90/MPI and openMP. It has been shown to achieve convergence for up to 16-nucleon problems with 2-body interactions [14] with present day computational resources. The MFDn code computes the lowest (typically 15) converged solutions, that is, the wavefunctions and eigenvalues, to the many-nucleon Schrödinger equation:

$$H \mid \psi_\alpha\rangle = E_\alpha \mid \psi_\alpha\rangle.$$

The nuclear wave functions satisfy the above equation so that they represent eigenstates $\Psi_\alpha >$ of the Hamiltonian operator H with the eigenvalues E_α as the corresponding energy. The building blocks of the MFDn code include the construction of the many-body Hamiltonian matrix for 2- or 3-body nuclear interactions, diagonalization of the Hamiltonian matrix, and computation of observables. At the end of a run, MFDn writes the nuclear wavefunctions to file (represented as coefficients of the basis space Slater determinants), and evaluates selected physical observables which can be compared to experimental data. The main computational challenge arises from the matrix size of the Hamiltonian operator. On currently available machines, the largest many-body basis dimension that MFDn can deal with for 3-body forces is 1 billion, and for 2-body forces is 10 to 15 billion [13]. In order to ensure we have provenance, several variables need to be recorded:

- System related information includes code name, code version number, compiler used, machine name, user name, and environment variables, etc.,
- Physics related variables such as numbers of protons and neutrons (Z,N), Hamiltonian inputs (2-body or 3-body interactions), number of observables to be computed, etc.,

- Truncation related variables such as the many-body basis space cutoff $N_{\max}$, the harmonic oscillator energy of the basis space $\hbar\omega$, number of Lanczos iterations, etc.

This information is identified by the team of MFDn developers, and it is necessary for rerunning the exact same experiment.

In addition to the provenance information, we also need to store the outputs, which includes a summary of the results, the one-body density matrices, the wavefunctions, and a description of the basis. We also need to log the data archive location for the output files. The form of the output for this code is currently as a set of binary or ASCII files whose size can range from kilobytes for the summary files to terabytes for the largest wavefunctions. The experiment may also have large size of input files associated with it, for example, ^{12}C using the 2-body potential may have input files of approximately 250 megabytes while a large 3-body run will have input files of about 33 gigabytes. For the 2-body interaction, experiments can run up to $N_{\max} = 18$, while with 3-body interactions we can only go to $N_{\max} = 8$ for the largest runs due to the memory limitation on current supercomputers.

In this paper we present a framework popularly called dedicated data management system designed to promote reproducible research in the area of nuclear physics.

4 Data Management System

A general DataBase Management System (DBMS) should perform the following functions:

- The inquiry or retrieval activity that accesses previously stored data in order to determine the recorded status of some real world entity or relationship,

- To update, which includes the original storage of data, its repeated modification as things change, and ultimately, its deletion from the system when the data is no longer needed.

Present day DBMSs place control of database development in the hands of database administrators (DBAs) who specialize in maintenance, creation and optimization of database schema and queries. SQL is the most popular query language for relational databases.

Even though SQL is very close to natural language, it is a technology that is best used by an expert. The objective of this study is to create a system, which superficially is more friendly to users, while underneath it is conversant with a formal DMS like SQL. The DMS we talk about in this paper is not comparable to the basic relational DMS with a dedicated storage engine. Instead, we use the basic DMS as a foundation to create a dedicated management system for this data.

4.1 MOTIVATION: Significance of Data Management System for Research Reproducibility

With the strong capabilites of tuple calculus, relational algebra, powerful optimizing techniques, and a variety of powerful engines capable of indexing and storing high volumes of data efficiently, the relational model of data storage is the most desirable choice. The primary data generated (section 3) is an ordinary ASCII file. This simplicity is introduced purposely as the files are created independently of the working of database schema. The back end of DMS can thus be fairly translated into conversion of plain text to object oriented data and then to relational data. In the absence of a middle stage (object oriented data), the back-end client code would have to create relational data and insert it directly into the database (tight coupling between database and client code). This would create complications in the client code and non-modularity of the software.

4.2 Challenges

In ensuring our DMS provides all the necessary information required for RR, we met with several challenges, discussed bellow.

4.2.1 Choice of Client-Server Coupling

There are two ways to record the provenance information from an application: a provenance tool could provide its own instrumentation library or routines that are then inserted into an application to collect provenance or, one could use a script to capture the experiment related data and generate a particular purpose output for later information recording. The first method is called tight coupling. It provides completeness and consistency of the resulting provenance, but can impose substantial performance overhead on the application and programmer. It requires that the DMS provides the Application Programming Interface (API) for MFDn code to use when inserting the experimental data to the database during the run. The second method is called loose coupling, which allows the MFDn code and the DMS to run separately. In loose coupling, the provenance information about the application is stored in a file during each run. This file contains all the information necessary to rerun the application as well as the result for future validation and analysis. This file can be in the form of xml or plain text. A specially developed parser reads this file and converts it into a readable format for the database.

In order to avoid the high application burden from the first method, we choose to separate the database connection from the original code. The library containing subroutines which parse and insert data is a part of a DMS. Loose coupling is preferred in this case due to its simplicity and superior modularity.

4.2.2 Large Size of Output Files

A wavefunction file for a run of MFDn (see section 3) can be as large as terabytes. This introduces high-performance

computing (HPC) challenges of storing and accessing those files. Database indexing and storage can become very slow for large databases. Handling a database with this amount of data is another topic of research not covered in this paper. Currently none of the output files are stored in the database; instead, they are stored in a shared directory structure, typically on the platform where they are generated. Large files (wavefunctions), as well as duplicates of the smaller files, are typically archived in a High Performance Storage System (HPSS). The strategy adopted in this case is to store the file path, rather than the files themselves. The provenance and log files are also stored as their paths. When the files are moved to different storage location, a script needs to run to update the database record with the new file path.

4.2.3 Ensuring Provenance

Provenance information associated with MFDn is described in 3, and the info file described in section 4.7 can efficiently store this information including input, environment variables and output of a run. This information may be crucially important and a distinguishing factor for future runs. This is especially true for a computational run considerably large in size. Besides the provenance information we discuss above, the log of each run is also needed for a reproducible run. We use another file to store this step by step log of our Python script generator so that a reproducible run can be generated based on this file. DMS stores the path to this file.

4.2.4 Ensuring Security

Database privileges are permissions granted by the database administrator to access the DMS. To a user viewing information online, only read privileges are granted; this reduces the risk of data corruptions. Permissions on the file system level are granted by storage authorities where the results of MFDn are stored. Files may be transferred to various locations over a period of time; care must be taken to update these locations. A web server separates the user from the actual results and code, in order to ensure a better scheme of authorizations. Some users may actually have complete access to code, data and results, while others may have access to none of this information, but still have access to the database content through the website. The server can allow or deny access to users using login authentication or based on IP address. We deploy the first method of authentication. This feature allows the DMS to have restricted access. For the case of MFDn, we encapsulate the parser, drop box, database server and web server at a common location, referred to from now on as DMS server, allow global access to website, and deny write access to database content. Parser and drop box locations are accessible only to the users of the DMS server.

4.3 Database Design

The structure of the data available is much like a tree structure. One run is the parent entity of many children entities. In relational databases, the process of normalization forces a

multivalued attribute to be decomposed from the parent table into a separate relation. This is the first normal form. Each run of MFDn has several multivalued attributes, e.g. Hamiltonian directories used, result files created etc. All these are in the form of separate tables, forming many children of a single parent table.

The metadata about runs gives rise to following entities:

(run, obdmefile, hamdir, resfile, smwffile)

The entity *run* identifies a experimental run. The next four entities reference the main *Run* table with multiple values for Hamiltonian directories used (*hamdir*), result files (*resfile*) , wavefunction files (*smwffile*), and one body density matrix files (*obdmefile*) produced. The multiple values for these files are primarily due to different values of various run parameters like $\hbar\omega$, the harmonic oscillator frequency in the calculation, as the same nucleus can be run using multiple values of these parameters.

For supporting HDF5, a child table is created to refer to the *RUN* table. The primary key is the dataset ID. Since HDF5 gives a numeric ID to each dataset, we can get wavefunction dimensions for each dataset and upload them for reference.

The database schema is demonstrated in Figure 1.

4.4 Workflow

There are two independent workflows (Figure 2) in this system. The front end workflow retrieves a record from the database and displays it on the webpage; HTTP requests are passed to the web server through a search form; a search result is then returned for display. In the back end, a script runs periodically to parse *mfdn.info* files and creates an instance of the structure 'run' for each *mfdn.info* file; the script then connects to the SQL server and performs a SQL query to upload the content of this instance.

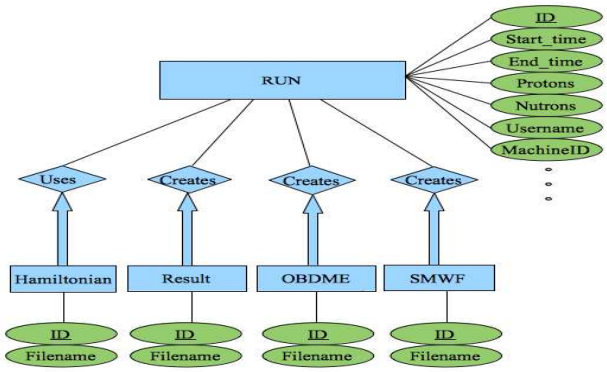


Figure 1. Simple Schema to Store Results

Figure 2. (1)Front End and (2) Back End Workflow

4.5 Software System Components

Figure 3 is the complete component based design for the system.

The dropbox and parser are parts of the back end component while the web server is a part of the front end component. Both the front end and the back end access the database, but the front end and back end components run independently of

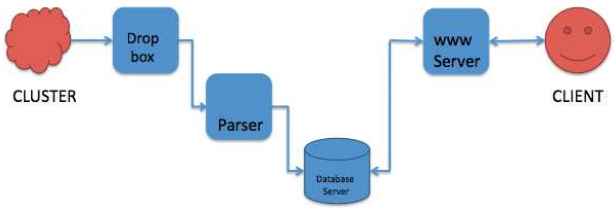


Figure 3. Component Design

each other. This ensures that in case of failure or malfunction of one, the other still works correctly. The process of modularization of these parts of the system is also facilitated by this design.

It is possible to either encapsulate 4 parts of the DMS at one location or keep them separate and

- DROP BOX: A location accessible for users of mfdn where each run delivers a file containing the report of the run (info file mentioned above),
- PARSER: This part of the design accommodates libraries which contain subroutines for parsing and querying. Parsing means to read each info file from the drop box and convert the textual data into a C structure. Querying means to insert each of the elements of this structure into their respective table,
- DATABASE: A relational database with schema described above,
- WEB SERVER: This part of the system deals with presentation of data in the database in a human readable format. It can be seen hereby that DMS forms an interface between complex parallel codes and post processing client.

There are four reasons to choose a web based system: increasing internet speeds and accessibility make web service the best option as far as data accessibility is concerned; enabling use of powerful dynamic scripting languages like PHP; enabling rendering of better look and feel and user friendliness; and highly efficient data manipulation with little network overhead and no data loss.

4.6 Support for HDF5 files

The recent development on the I/O side of MFDn is on the output in HDF5 format [10] which is a versatile data model that can represent very complex data objects and a wide variety of metadata. The main reason is that sequential I/O is a major bottleneck in parallel codes, while HDF5 is a portable, human-readable and self-describing format that tackles the bottleneck of I/O time taken for large output datasets. In MFDn, the construction of the Hamiltonian matrix and diagonalization are the most costly, but sequential I/O is becoming a bottleneck when writing the wavefunctions out at the end of the experimental run. Currently parallel I/O (both MPI I/O and HDF5) are supported in MFDn code. Figure 4 demonstrates how well the HDF5 write outperforms sequential binary write. By using HDF5, an extra benefit can be obtained such as the flexibility on adding data anytime and anywhere in the file which is useful for certain properties which are calculated later, and HDF5 simplifies the file structure to include only two major types of objects: Datasets and Groups. The MFDn HDF5 output contains various useful details such as dimensions of eigen vectors calculated and values for corresponding eigen vectors. It also stores information about partition of wavefunctions among nodes and processors. This information can be accessed by querying the HDF5 file using C or Fortran API's provided by the HDF group [2]. This task

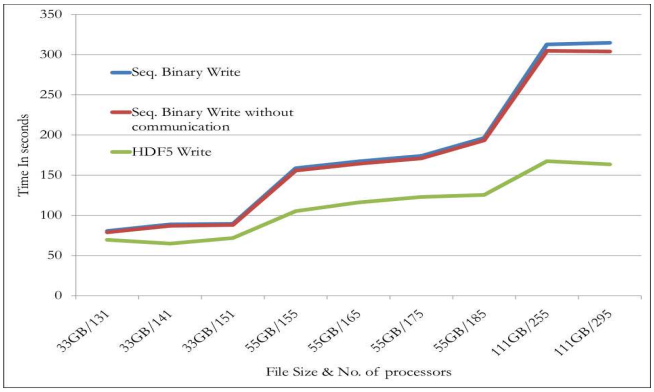


Figure 4. Comparison of Sequential Binary and Parallel HDF5 Write Operations

is simplified by the DMS through parsing the HDF5 file and inserting these details about the output wavefunctions into the database. It is impractical to transfer terabytes of data to the local server for retrieving the header information. Therefore, the decision is made to use HDF5 dump functionality that creates header file which is much smaller than actual output on the remote supercomputer and transfer the header information back to the local server. This part of the DMS can be explained with the following pseudo code:

```
1.Read /project folder for new hdf5 files.
2.Copy files in /project/temp folder
3.While /project/temp is not empty
4.Copy oldest file into source folder
5.Dump xml from hdf5 file
6.Parse xml file by querying for vector dimensions and
corresponding dataset names
7.Insert each variable into database
8. If 5,6 and 7 succeed then move hdf5 file into processed/
9. Else move hdf5 file to error/
10. end while
11. Stop
```

4.7 Implementation

In order to generate the necessary input data for the DMS, an interactive python script is utilized to set up a run of MFDn at facilities. such as National Energy Research Scientific Computing Center (NERSC) and Oak Ridge National Laboratory (ORNL). This python script also generates a provenance file, which captures all interactive input from the user to the python script. At the end of each MFDn run, a file called *mfdn.info* is generated and concatenated with a timestamp. The structure of this file is basically custom made for this parser. It contains metadata about each run in the form of *(key, value)* pairs, where the first column represents the attributes in the database and the second column stores the values. The attributes are determined by the physicists who develop and use the MFDn code for their nuclear physics research. The file is transferred to the drop box by the person responsible for running the client code. The following snippet shows the partial info file parsed by parser:

```
START_DATE  Sat May 1 02:06:50 CDT 2010
 username    jvary
 machineID   nuclear_server
 jobID       NONE
 rundir      /home/jvary/mfdn_v13_rev254
 hamdir
/project/hamilops;
/project/hamilops/Minnesota
 2B_potential     Minnesota
 3B_potential     NONE
 4B_potential     NONE
 renormalised     NONE
 ext_field        HO hw10
 Z           0
 N           8
 nshell_min_Z           0
 nshell_min_N           1
 nshell_max_Z           0
 nshell_max_N           8
```

The back end workflow in Figure 2 (2) can be better understood by the following pseudo code:

```
1.Read /project folder for new info files.
2.Copy  info files in /project/temp folder
3.While /project/temp/ is not empty
4.Copy oldest file into source  folder
5.Parse info file into structure 'runinfo'
```

```
6.Insert each variable in 'runinfo' into database
7. If 5 and 6 succeed then move info file into /project/processed/
8. Else move info file to /project/error/
9. end while
10.Stop
```

Here the directories sit on the local server (*nuclear.physics.iastate.edu*) where the database is.

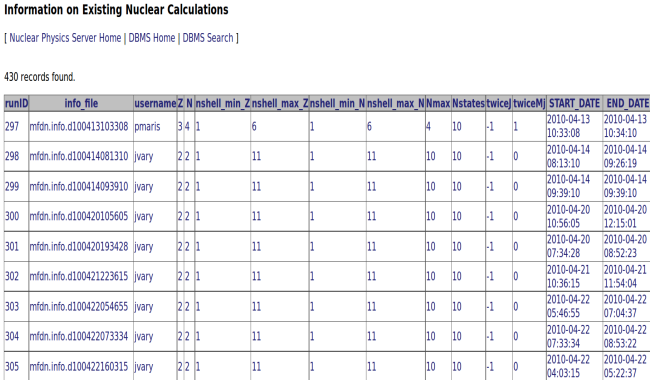
The front end in Figure 2 (1) is basically a web service which continuously responds to client requests as per the type of request. The following tasks are provided in the front end:

```
1.Search run
2.View run details
3.Modify run details
```

5 A Case Study

The DMS for MFDn provides a simple and scalable platform for the organization of, and access to, research results located across multiple storage locations. For scientific and research purposes, this facilitates the aggregation and analysis of data produced by any group contributing to the DMS. The DMS is also an excellent educational resource. In a classroom setting, it is impractical to perform a detailed nuclear structure calculation; however, the DMS provides a simple and user-friendly method to access and compare nuclear structure calculations performed with various methods, levels of accuracy, and nuclear potential interactions, among others. This makes it possible to have detailed discussions about advanced nuclear structure calculations without becoming mired in details, i.e. which methods are best under different circumstances, the benefits and drawbacks of different potential interactions, or the effect that the size of calculations and where one decides to truncate their approximations have on final results.

The DMS is online at *http://nuclear.physics.iastate.edu/info*. Options such as view all the runs, and search for the runs are provided. Figure 5 lists all existing runs from database. Eeach entry represents one run with brief information about the run, and is hyperlinked with detailed run information as in Figure 6



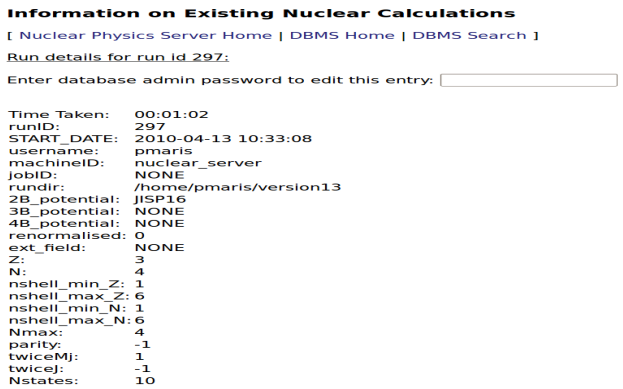
Information on Existing Nuclear Calculations

[Nuclear Physics Server Home | DBMS Home | DBMS Search]

430 records found.

runID	info_file	username	Z	N	nshell_min_Z	nshell_max_Z	nshell_min_N	nshell_max_N	Nmax	Nstates	twicej	twiceMj	START_DATE	END_DATE
297	mfdn.info.d100413103308	pmaris	3	4	1	6	1	6	4	10	-1	1	2010-04-13 10:33:08	2010-04-13 10:34:10
298	mfdn.info.d100414081310	jvary	2	2	1	11	1	11	10	10	-1	0	2010-04-14 08:13:10	2010-04-14 09:26:19
299	mfdn.info.d100414093910	jvary	2	2	1	11	1	11	10	10	-1	0	2010-04-14 09:39:10	2010-04-14 09:39:10
300	mfdn.info.d100420105605	jvary	2	2	1	11	1	11	10	10	-1	0	2010-04-20 10:56:05	2010-04-20 12:15:01
301	mfdn.info.d100420193428	jvary	2	2	1	11	1	11	10	10	-1	0	2010-04-20 07:34:28	2010-04-20 08:52:23
302	mfdn.info.d100421223615	jvary	2	2	1	11	1	11	10	10	-1	0	2010-04-21 10:36:15	2010-04-21 11:54:04
303	mfdn.info.d100422054655	jvary	2	2	1	11	1	11	10	10	-1	0	2010-04-22 05:46:55	2010-04-22 07:04:37
304	mfdn.info.d100422073334	jvary	2	2	1	11	1	11	10	10	-1	0	2010-04-22 07:33:34	2010-04-22 08:53:22
305	mfdn.info.d100422160315	jvary	2	2	1	11	1	11	10	10	-1	0	2010-04-22 04:03:15	2010-04-22 05:22:37

Figure 5. Information on Existing Nuclear Calculations



Information on Existing Nuclear Calculations

[Nuclear Physics Server Home | DBMS Home | DBMS Search]

Run details for run id 297:

Enter database admin password to edit this entry:

Time Taken:	00:01:02
runID:	297
START_DATE:	2010-04-13 10:33:08
username:	pmaris
machineID:	nuclear_server
jobID:	NONE
rundir:	/home/pmaris/version13
2B_potential:	JISP16
3B_potential:	NONE
4B_potential:	NONE
renormalised:	0
ext_field:	NONE
Z:	3
N:	4
nshell_min_Z:	1
nshell_max_Z:	6
nshell_min_N:	1
nshell_max_N:	6
Nmax:	4
parity:	-1
twiceMj:	1
twicej:	-1
Nstates:	10

Figure 6. Detailed information on Existing Nuclear Calculations

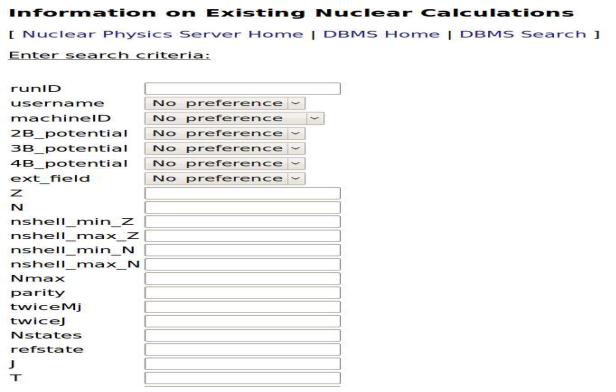
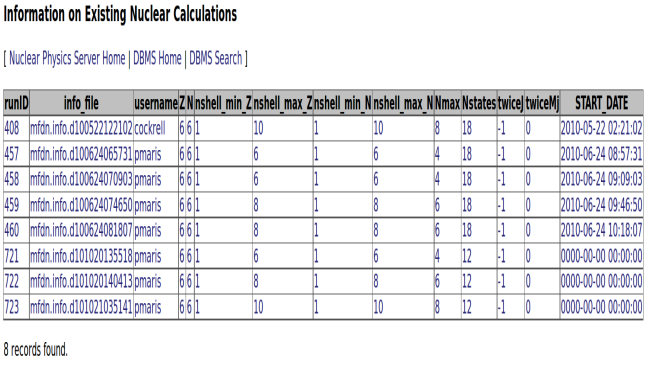


Figure 7. Search on Existing Nuclear Calculations



Information on Existing Nuclear Calculations

[Nuclear Physics Server Home | DBMS Home | DBMS Search]

runID	info_file	username	Z	N	nshell_min_Z	nshell_max_Z	nshell_min_N	nshell_max_N	Nmax	Nstates	twiceJ	twiceMj	START_DATE
408	mfdn.info.d100522122102	cockrell	6	6	1	10	1	10	8	18	-1	0	2010-05-22 02:21:02
457	mfdn.info.d100624065731	pmaris	6	6	1	6	1	6	4	18	-1	0	2010-06-24 08:57:31
458	mfdn.info.d100624070903	pmaris	6	6	1	6	1	6	4	18	-1	0	2010-06-24 09:09:03
459	mfdn.info.d100624074650	pmaris	6	6	1	8	1	8	6	18	-1	0	2010-06-24 09:46:50
460	mfdn.info.d100624081807	pmaris	6	6	1	8	1	8	6	18	-1	0	2010-06-24 10:18:07
721	mfdn.info.d101020135518	pmaris	6	6	1	6	1	6	4	12	-1	0	0000-00-00 00:00:00
722	mfdn.info.d101020140413	pmaris	6	6	1	8	1	8	6	12	-1	0	0000-00-00 00:00:00
723	mfdn.info.d101021035141	pmaris	6	6	1	10	1	10	8	12	-1	0	0000-00-00 00:00:00

8 records found.

Figure 8. Search Result on Existing Nuclear Calculations

6 Related Work

The National Nuclear Data Center (NNDC) [1] has an online website for experimental nuclear structure data which contains two databases. Evaluated Nuclear Structure Data File (ENSDF) contains evaluated nuclear structure and decay information for over 3000 nuclides in a standard format. Experimental Unevaluated Nuclear Data List (XUNDL) contains compiled nuclear structure data in the "ENSDF" format, and the XUNDL database contains experimental data compiled from over 2300 recent nuclear structure papers. An international network of evaluators contributes to the database, which is maintained by the NNDC at Brookhaven National Laboratory (BNL). For each nuclide, all known experimental data used to deduce nuclear structure information are included. These databases are mainly for nuclear data sharing, contain extensive listings of the resource literature but not intended to provide detailed provenance information.

7 Conclusion and Future Work

In this work, a data management system is built for the *ab-initio* Nuclear Physics code, MFDn. The numerical results and suitable provenance information such as scripts, compilers, and hardware info are retrievable through the DMS, so that an expert researcher could reproduce the data. HDF5 files are also supported as well as binary output format. In the future, we'd like to apply this system to other CI Nuclear Physics codes such as Bigstick [4] and NuShellX [3, 5]. A common data management system API will be proposed to the community to gain consensus.

This can also be regarded as a first necessary step towards workflow provenance of an experiment. Specifically, a dedicated system such as Kepler [11] can be used along with continual update of the database as a robust and long term research method. The case studied in this paper is a parallel algorithm which runs on high performance processors; it is important to investigate global reproducibility of such a code. Further work is needed to insure the means of availability and operability of such codes for any person who may be studying research papers based on them.

8 Acknowledgement

This work was supported in part by Iowa State University under the contract DE-AC02-07CH11358 with the U.S. Department of Energy, by the U.S. Department of Energy under the grants DE-FC02-09ER41582 (UNEDF SciDAC-2) and DE-FG02-87ER40371 (Division of Nuclear Physics), by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract number DE-AC02-05CH11231, and in part by the National Science Foundation grant NSF/OCI – 0904782.

References

[1] National Nuclear Data Center Website. `http://www.nndc.bnl.gov/`, Nov 2010.

[2] The HDF Group. `http://www.hdfgroup.org/`, Nov 2010.

[3] W. D. M. Rae. `http://knollhouse.eu/NuShellX.aspx`, Nov 2010.

[4] W.E. Ormand, C. W.Johnson, and P.G. Krastev, private communication, Nov 2010.

[5] B. A. Brown and W. D.M. Rae and E. McDonald and M. Horoi. NuShellX@MSU. `http://www.nscl.msu.edu/~brown/resources/resources.html`, Nov 2010.

[6] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science and Engineering*, 10:11–21, 2008.

[7] P. Groth, S. Miles, and L. Moreau. A model of process documentation to determine provenance in mash-ups. *ACM Trans. Internet Technol.*, 9:3:1–3:31, February 2009.

[8] J. Kim, E. Deelman, Y. Gil, G. Mehta, and V. Ratnakar. Provenance trails in the wings-pegasus system. *Concurr. Comput. : Pract. Exper.*, 20:587–597, April 2008.

[9] J. Kovacevic. How to encourage and publish reproducible research. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages IV–1273 – IV–1276, 2007.

[10] N. Laghave, M. Sosonkina, P. Maris, and J. P. Vary. Benefits of parallel i/o in ab initio nuclear physics calculations. In G. Allen, J. Nabrzyski, E. Seidel, G. D. van Albada, J. Dongarra, and P. M. A. Sloot, editors, *Computational Science - ICCS 2009, 9th International Conference, Baton Rouge, LA, USA, May 25-27, 2009, Proceedings, Part I*, volume 5544 of *Lecture Notes in Computer Science*, pages 84–93. Springer, 2009.

[11] B. Ludascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. Scientific workflow management and the kepler system. In *Concurr. Comput. : Pract. Exper*, page 2006, 2005.

[12] P. Maris, A. M. Shirokov, and J. P. Vary. Ab initio nuclear structure simulations: The speculative ^{14}F nucleus. *Phys. Rev. C*, 81(2):021301, Feb 2010.

[13] P. Maris, M. Sosonkina, J. P. Vary, E. Ng, and C. Yang. Scaling of ab-initio nuclear physics calculations on multicore computer architectures. *Procedia Computer Science*, 1(1):97 – 106, 2010. ICCS 2010.

[14] P. Maris, J. P. Vary, and A. M. Shirokov. Ab initio no-core full configuration calculations of light nuclei. *Phys. Rev. C*, 79(1):014308, Jan 2009.

[15] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. Seltzer. Provenance-aware storage systems. In *Proceedings of the annual conference on USENIX '06 Annual Technical Conference*, pages 4–4, Berkeley, CA, USA, 2006. USENIX Association.

[16] P. Navrátil, V. G. Gueorguiev, J. P. Vary, W. E. Ormand, and A. Nogga. Structure of $A = 10 - -13$ Nuclei with Two- Plus Three-Nucleon Interactions from Chiral Effective Field Theory. *Phys. Rev. Lett.*, 99(4):042501, Jul 2007.

[17] P. Navrátil, J. P. Vary, and B. R. Barrett. Properties of ^{12}C in the Ab Initio Nuclear Shell Model. *Phys. Rev. Lett.*, 84(25):5728–5731, Jun 2000.

[18] P. Sternberg, E. G. Ng, C. Yang, P. Maris, J. P. Vary, M. Sosonkina, and H. V. Le. Accelerating configuration interaction calculations for nuclear structure. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, SC '08, pages 15:1–15:12, Piscataway, NJ, USA, 2008. IEEE Press.